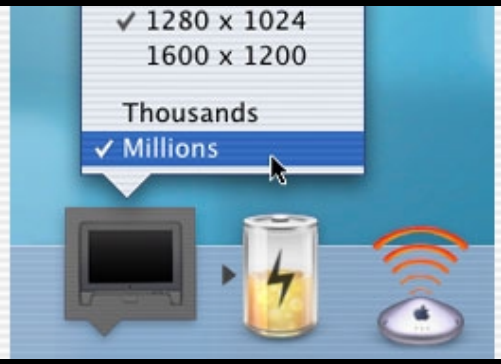
Session 506

Mac OS X Features in Java Applications



Peter A. Steinauer
Engineering Manager, Java Platform Classes

Goals

- Minimal changes to 100% pure applications
- APIs that blend well with Sun's
- Applications that FEEL like Mac OS X apps

Overview

- The basics
- Simple tweaks
- More complex features
- Packaging

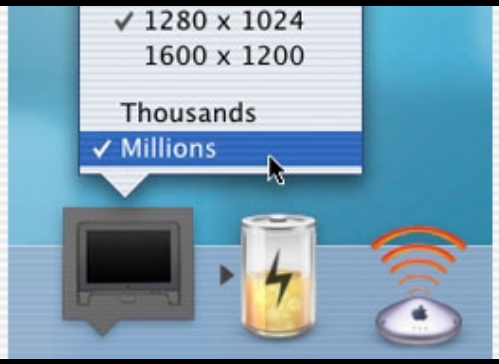
The Basics

Cross Platform Changes

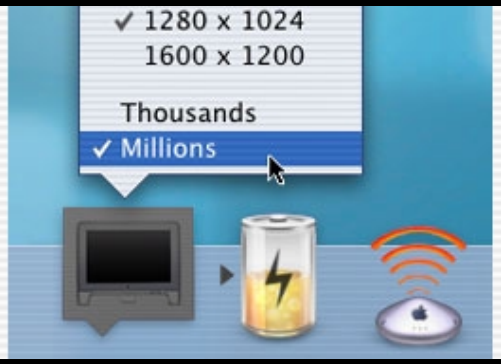
The Basics

- Clean 100% Pure Swing Implementation
- Coding style
  - Modularity
  - Eye for platform differences
- Isolated native methods
- Launchability—the Manifest

Demo



Simple Tweaks



Distinguishing Platforms

- Distinguishing Platforms—os.name
  - ‘Mac OS’—Mac OS 9 (MRJ 2.2.x)
  - ‘Mac OS X’—Mac OS X
- Distinguishing Features—mrj.version
  - 2.2.x—Mac OS 9 Java (MRJ 2.2.x)
  - 3.0—Mac OS X Java
  - 3.1—Mac OS X Java Update

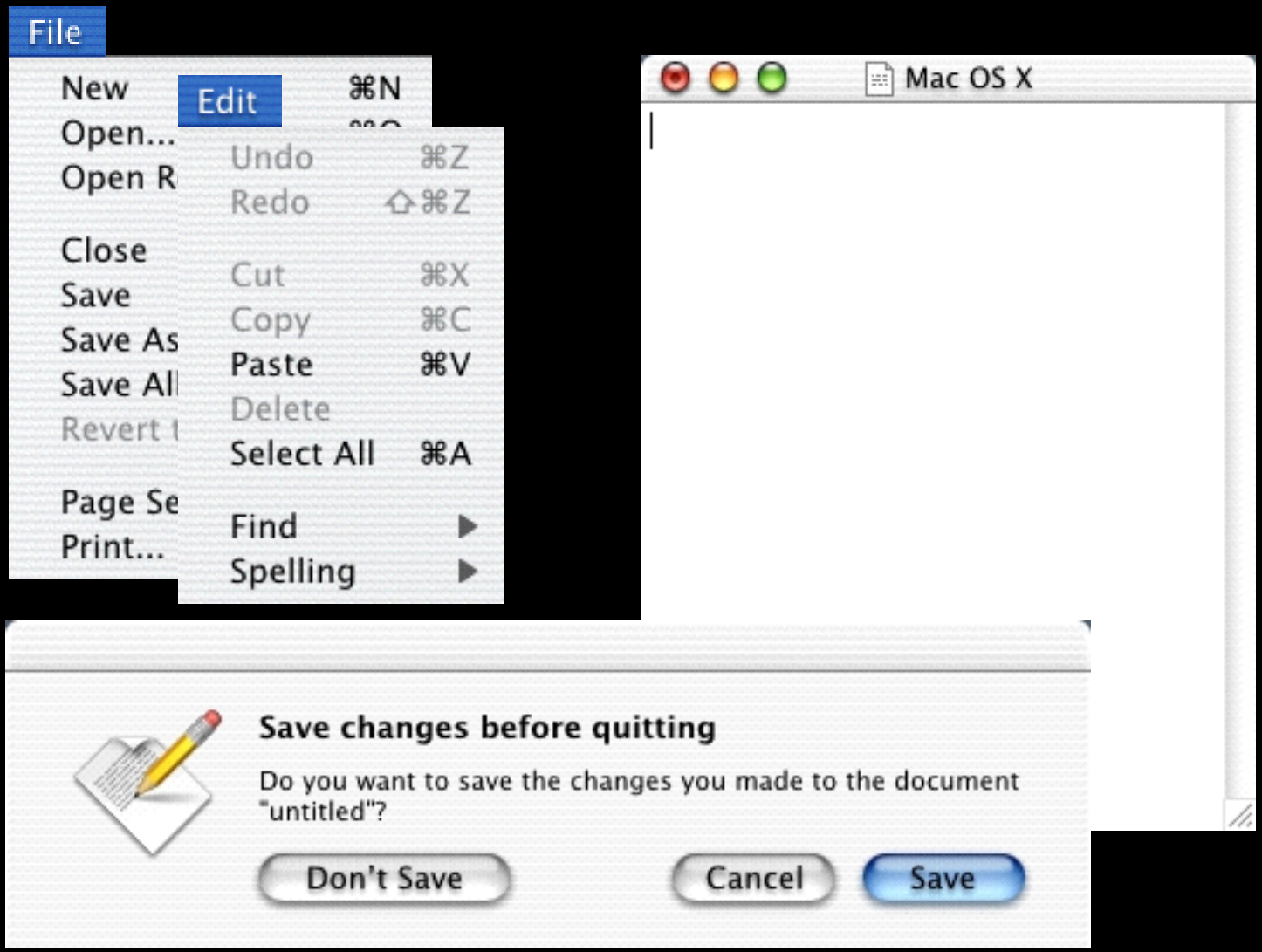
Choosing a UI Look and Feel

```
try {
    UIManager.setLookAndFeel(
        UIManager.getSystemLookAndFeelClassName());
    SwingUtilities.updateComponentTreeUI( myJFrame );
}
catch( Exception e ) {
    System.out.println( “We’re HOSED” );
}
```

Menu Definition and Layout

- Call setJMenuBar() on JFrames and JDialogs
  - Can't have a heavyweight menu bar already associated
  - Screen MenuBars currently don't work on JDialogs
- To enable set 'com.apple.macos.useScreenMenuBar=true' property

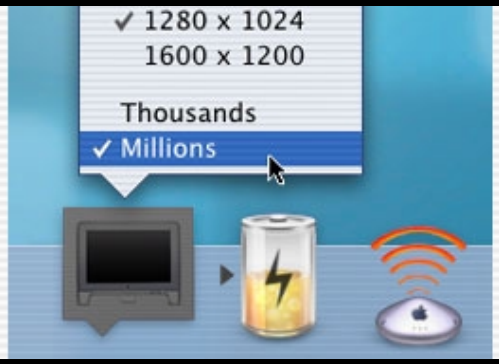
Mac HI Basics



Mac HI Basics

- Menus
  - Layout
  - Accelerators
- Windows/Dialogs
  - Layout
  - Behavior

Demo



More Complex Features

- Drag and Drop
- MRJ Toolkit
- Handling Basic Apple Events
- File System Manipulation
- JDirect 3/Carbon Locking

Drag and Drop

- java.awt.dnd / java.awt.datatransfer
  - Good 90% Solution
- com.apple.mrj.dnd / com.apple.mrj.datatransfer
  - Multiple non-file drag items
  - Conversion between Mac OS data types and Java

MRJToolkit

- MRJOSType
- MRJFileUtils
  - Type/Creator
  - Locating Apps/Special Folders
- MRJApplicationUtils
  - Installing Event Handlers

Handling Basic AppleEvents

```
import java.io.File;
import com.apple.mrj.MRJQuitHandler;
import com.apple.mrj.MRJApplicationUtils;

public class MyClass implements MRJOpenFileHandler
{
    public MyClass {
        if( Configuration.getPlatform() ==
                    Configuration.PLATFORM_MACOS )
            MRJApplicationUtils.registerOpenFileHandler( this );
    }

    public void handleOpenFile( File openMe )
        { ... }
}
```

File System Access—PROPOSAL

- MacFile—Mac OS X Specific File Attributes
  - Type/Creator (Deprecated)
  - Finder Attributes (App, package, hidden, etc.)
  - more…
- MacApplication—Launch/Finder Assistance
  - Document/App launching
  - MIME Type Mapping/Application association
- Where to get it

  http://www.apple.com/developer/java

MacFile API

```
public class MacFile {
        public boolean isAlias()
        public boolean isApplication()
        public boolean isDirectory()
        public boolean isFile()
        public boolean isPackage()
        public String getCanonicalPath()
        public boolean isVisibleInFinder()
        public boolean isInPackage()
        public MacApplication getApplication()
        public Image getIcon(int pixelsPerSide, int bitsPerPixel)

        ....
}
```

MacApplication API

```
public class MacApplication {
        public File getFile()
        public void open()
        public void openDocument(File document)
        public void openDocuments(File[] documents)
        public boolean acceptsFile(File file)
        public String getLocalizedApplicationName()
        public int getVersion()
        public File getExecutableFile()

        ...
{
```

JDirect 3

```
// JDirect 3 example for Mac OS X
import com.apple.mrj.jdirect.Linker;

public class Prime {
        // The call to new Linker() installs the native methods
        static Object linkage = new Linker(Prime.class);

        // This string tells JDirect where to find the native code
        static final String JDirect_MacOSX = "prime.dylib";

        public static native long ComputePrime(short n);
}
```

JDirect 3 (Cont.)

```
import com.apple.mrj.jdirect.Linker;

public class MemoryFunctions {

    private static Object linkage
        = new Linker(MemoryFunctions.class);

    private static final String JDirect_MacOSX
        = "/System/Library/Frameworks/CoreServices.framework/CoreServices";

        // Ptr NewPtr(Size size);
        public static native int NewPtr(int size);
}
```

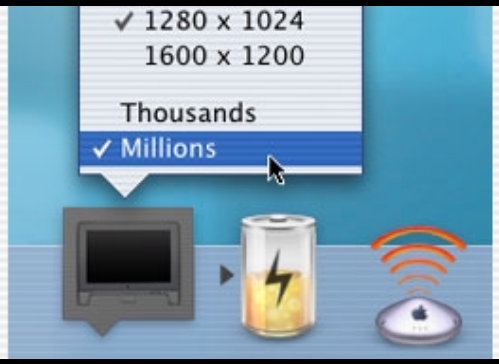
Carbon (Toolbox) Lock

```
import com.apple.mrj.macos.carbon.CarbonLock;

try {
        CarbonLock.acquire();

        <Carbon call here>
}
finally {
        CarbonLock.release();
}
```

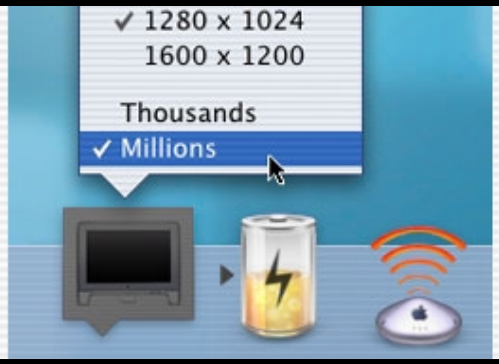
Demo



Packaging

- Basics
- Mac OS X additions
- WebStart

Demo



Other Resources

- Mac OS X HI Guidelines

  http://developer.apple.com/techpubs/macosx/SystemOverview/AquaHIGuidelines/index.html

- Mac OS X Java

  http://developer.apple.com/java

- Sample Code
- Java Update

  http://connect.apple.com/

Roadmap

507 QuickTime for Java: QuickTime in Java on Mac OS X	Civic **Thurs., 2:00pm**
508 Java Graphics: AWT Graphics, Java2D, Performance	Civic **Thurs., 3:30pm**
509 Java Security: Applet Security, J2SE Security, JSSE	Room J2 **Fri., 9:00am**
FF117 Feedback Forum: Java	Room J1 **Fri., 10:30am**

Who to Contact

Alan Samuel
Java Technology Manager
Apple Worldwide Developer Relations
bluker1@apple.com

Q&A

http://developer.apple.com/events/urls.html

WWDC
Worldwide Developers Conference 2001